Lab 8 (2%)

MongoDB: updating documents

Learning outcomes

Upon successful completion of this workshop, you will have demonstrated the abilities to:

1. Updating existing documents
2. Removing documents from a collection.
3. Group work acknowledgment

We, Divnoor Bhandohal and Sophia Huynh, declare that the attached assignment is our own work in accordance with the Seneca Academic Policy. No part of this assignment has been copied manually or electronically from any other source (including web sites) **or distributed to other students.**

Specify below what each member has done towards the completion of this work:

Name Task(s)

1- Divnoor Bhandohal 1 – 8

2- Sophia Huynh 9 - 16

3-

Instructions

Create a new database lab8 and run the following insert commands to insert the documents in the products collection:

db.products.insert(

{ code: "11QER/31", desc: "Power painter, 15 psi., 3-nozzle", indate:ISODate("2011-11-03"), qoh: 100, min: 5, price:109.99, colors: ["black", "red"], size: { l: 24, w: 19.99, h:14.99, uom: "cm" }, package: {weight: 1.79, uom: "kg"}}

)

db.products.insertMany([

{ code: "13-Q2/P2", desc: "7.25-in. pwr. saw blade", indate:ISODate("2011-12-13"), qoh: 32, min: 15, price:14.99, size: { w: 7.25, h:0.25, uom: "in" }},

{ code: "14-Q1/L3", desc: "9.00-in. pwr. saw blade", indate:ISODate("2011-11-13"), qoh: 18, min: 12, price:17.49, colors: ["Metallic"], size: { l: 9, uom: "in" }, package: {weight: 0.6, uom: "lbs"}},

{ code: "1546-QQ2", desc: "Hrd. cloth, 1/4-in., 2x50", indate:ISODate("2012-01-15"), qoh: 15, min: 8, price:39.95, colors: ["silver"], size: { l: 50, h: 2, uom: "ft" }, package: {weight: 19.5, uom: "lbs"}},

{ code: "1558-QW1", desc: "Hrd. cloth, 1/2-in., 3x50", indate:ISODate("2012-01-15"), qoh: 23, min: 5, price:43.99, size: { l: 50, h: 4, uom: "ft" }, package: {weight: 40, uom: "lbs"}}

])

**Updating documents**

1. Document replacement
   1. Find product 11QER/31 by code and assign it to a variable prod.
   2. Add the key named reviews to prod object as an array of reviews containing one review with the following key/value pairs:

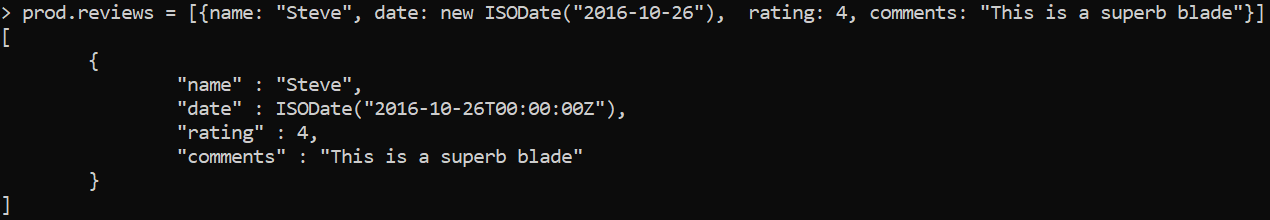
name: steve,

date: October 26, 2016 saved as date object,

rating: 4 ,

comments: “This is a superb blade.”

prod.reviews = [{name: "Steve", date: new ISODate("2016-10-26"), rating: 4, comments: "This is a superb blade"}]



* 1. Then replace the product database’s version using replaceOne() function.

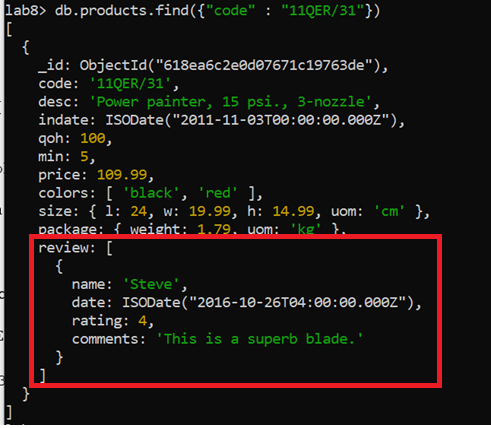
db.products.replaceOne({"code" : "11QER/31"}, prod)



* 1. Show product 11QER/31. The document schema should show review as an array of comments.



Output:

****

1. Update the colors attribute for product 11QER/31 to contain black and white. Show product 11QER/31 after the update.

db.products.update({"code" : "11QER/31"}, {$set:{"colors":["black","white"]}})

****

****

Output:

****

**Using modifiers**

**Use the $inc modifier**

1. The supply management has ordered 100 new items of product 11QER/31.
   1. Use find() and show product 11QER/31.



* 1. Use the “$inc” modifier to increase the quantity on hand (qoh) of product 11QER/31 by 100.

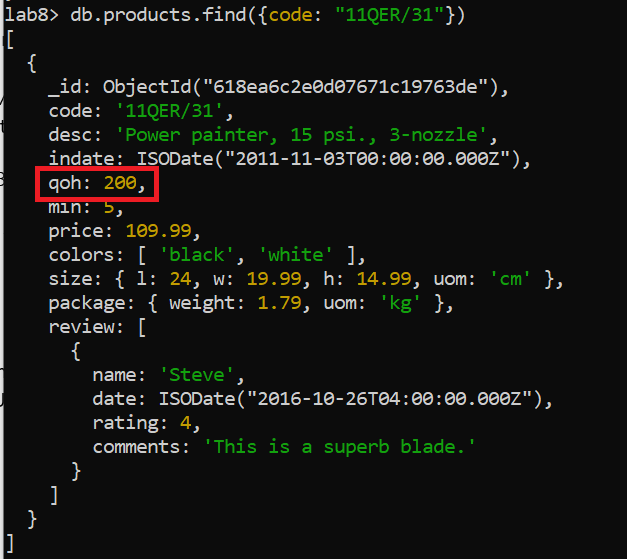
db.products.update({"code" : "11QER/31"}, {$inc: {qoh:100}})



* 1. Use find() and show product 11QER/31 after the update.



Output: qoh should be equal 200.



**Use the $set modifier**

1. The supply management has decided to change the minimum threshold to reorder a product.
   1. Use find() and show product 11QER/31.



* 1. Use the “$set” modifier to set the minimum quantity threshold to reorder for product 11QER/31 at 20.

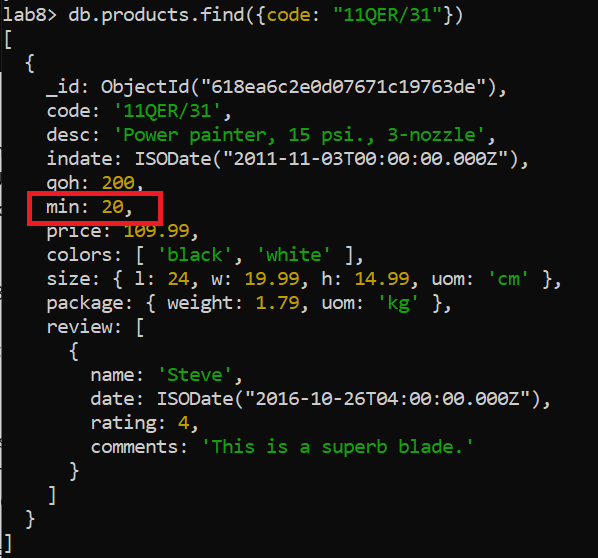
db.products.update({"code" : "11QER/31"}, {$set:{"min": 20}})



* 1. Use find() and show product 11QER/31 after the update.



Output: min changed from 5 to 20.



1. The package’s weight of product 11QER/31 has to change to 2.5kg.
   1. show product 11QER/31.



* 1. Use the “$set” modifier to set the package weight at 2.5kg.

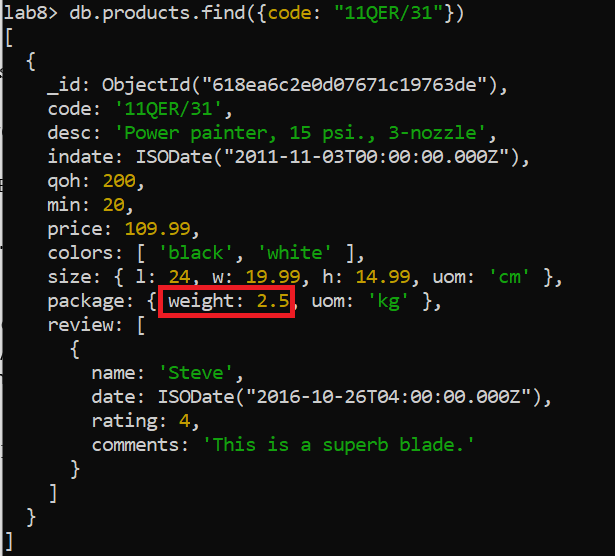
db.products.update({"code" : "11QER/31"}, {$set:{"package.weight": 2.5}})



* 1. show product 11QER/31 after the update.

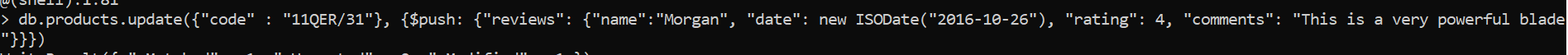


Output:

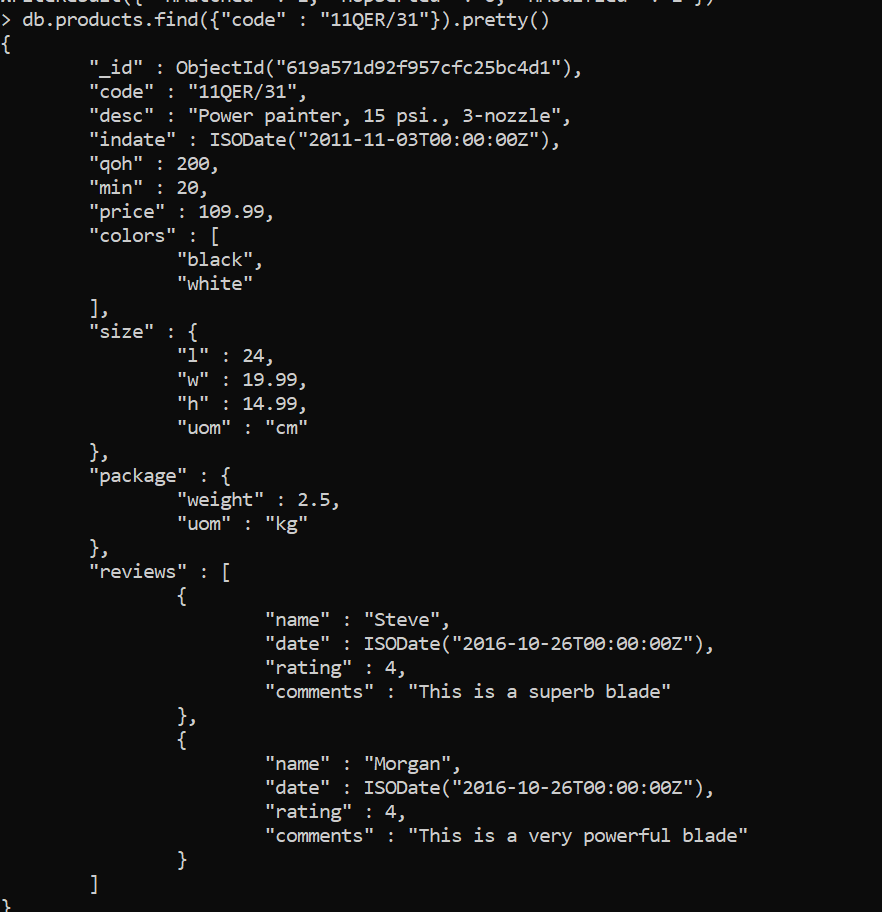


1. **Add elements to an array** 
   1. Use "$push" operator to add a review at the end of the reviews array for product 11QER/31. The review to add contains the following key/value pairs: name: morgan, date: October 26, 2016 saved as date object, rating: 4, comments: “This is a good blade.”

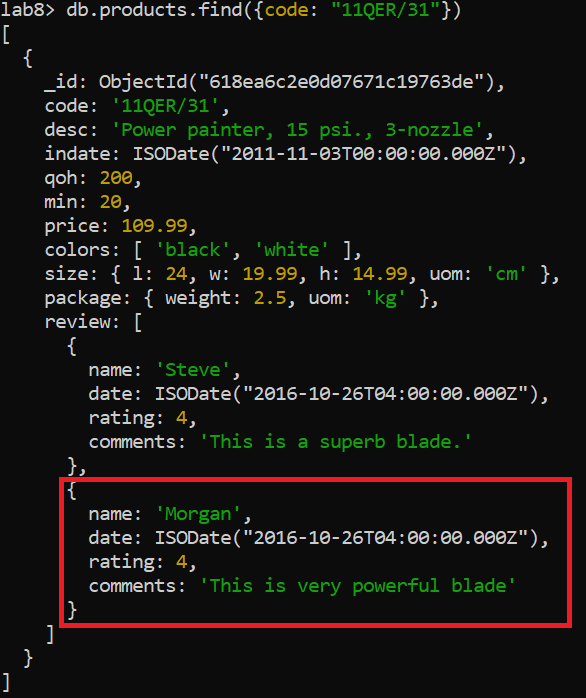
db.products.update({"code" : "11QER/31"}, {$push: {"reviews": {"name":"Morgan", "date": new ISODate("2016-10-26"), "rating": 4, "comments": "This is a very powerful blade"}}})



* 1. show product 11QER/31 after the update.



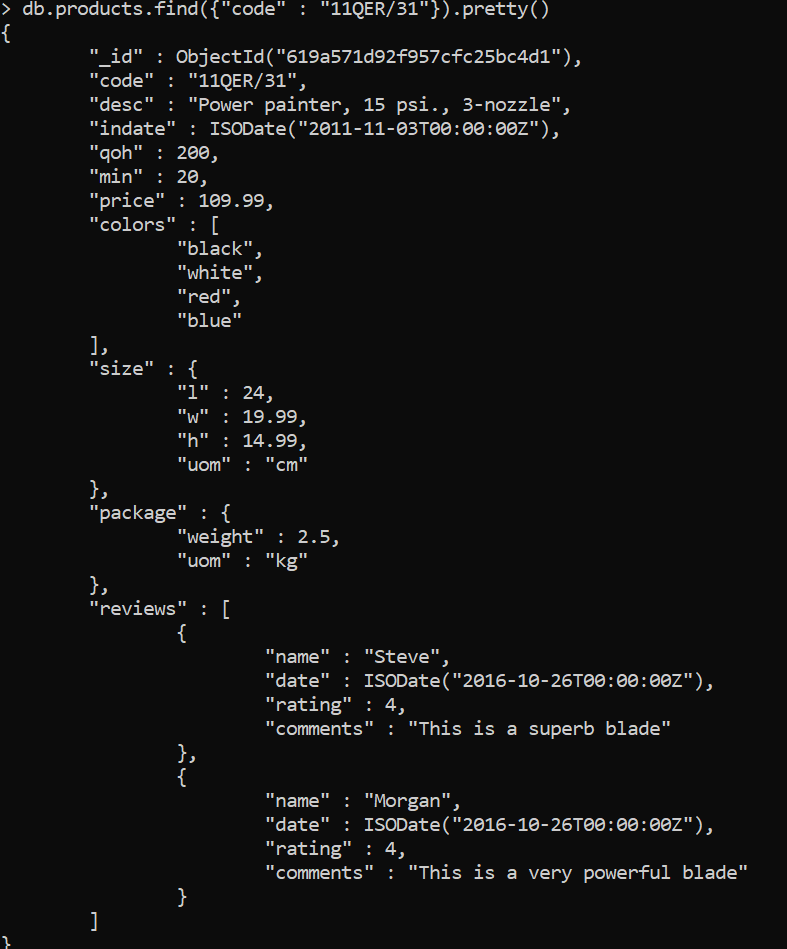
Output:



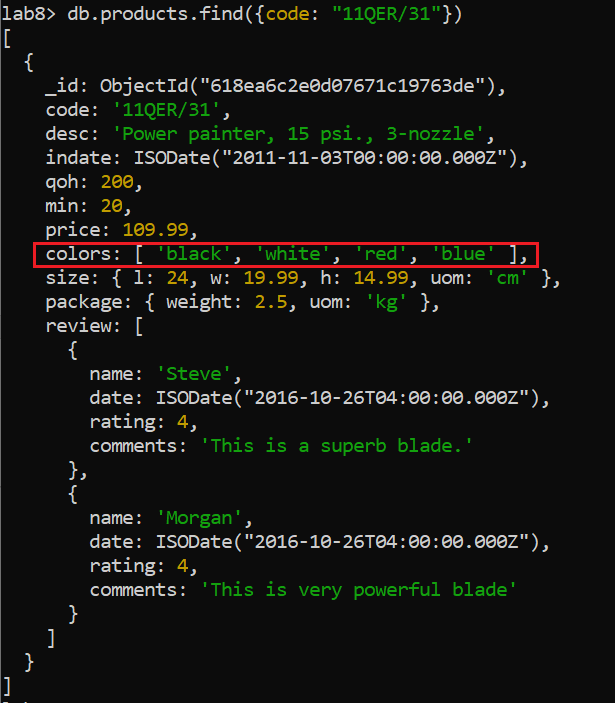
1. **Add multiple elements to an array** 
   1. Use $push and $each operators to push multiple colors to product 11QER/31 in one operation. Add the colors red and blue.

db.products.update({"code" : "11QER/31"}, {$push: {colors:{ $each: ["red","blue"]}}})

* 1. show product 11QER/31 after the change.

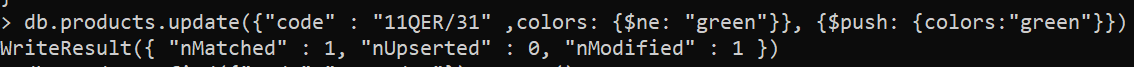


Output:

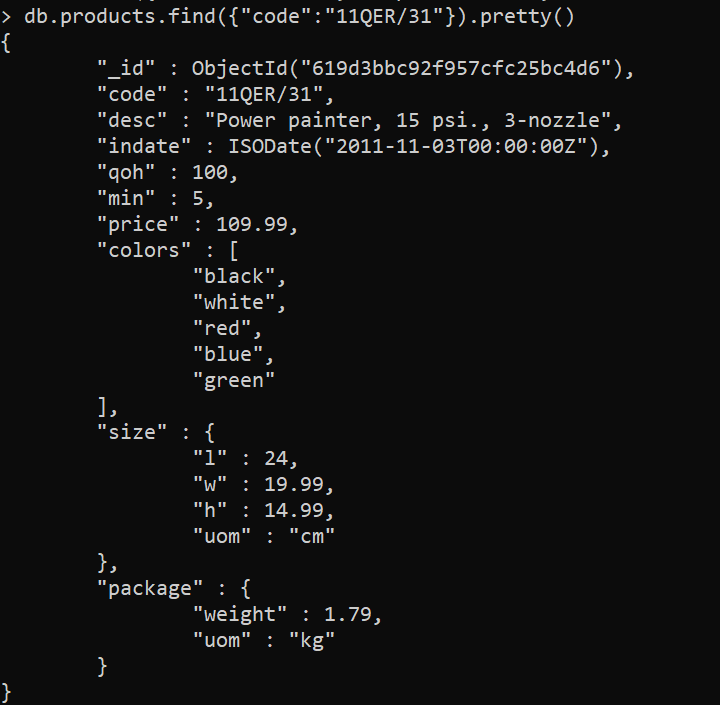


1. **Add element to an array if they do not exist.** 
   1. For product 11QER/31, add the color green to the colors array if it does not exist. The criterion to update should check the code and whether green is not in the array. Use $push and $ne operators.

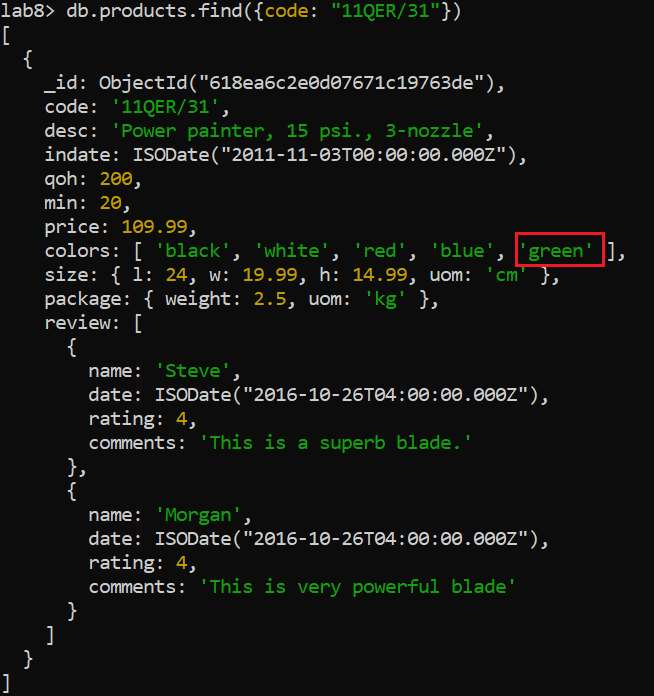
db.products.update({"code" : "11QER/31" ,colors: {$ne: "green"}}, {$push: {colors:"green"}})



* 1. Show product 11QER/31 after the changes.



Output:



* 1. Use the same command to add the color black, to the colors array. Make sure the color black is not added to the colors array.

db.products.update({"code" : "11QER/31" ,colors: {$ne: "black"}}, {$push: {colors:"black"}})



Output:

{

acknowledged: true,

insertedId: null,

matchedCount: 0,

modifiedCount: 0,

upsertedCount: 0

}

1. **Avoid adding duplicates to an array with $addToSet operator** 
   1. Use $addToSet operator to add “black” to the colors array for product 11QER/31. This product already has the black color in the list. The operation should not add duplicate values in the colors array.

db.products.update({code: "11QER/31"},{$addToSet: {colors: 'black'}});



* 1. Show product 11QER/31 after executing the command.



Output:

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 0,

upsertedCount: 0

}

1. **Add multiple elements and avoid duplicates.** 
   1. use "$addToSet" in conjunction with "$each" to add the colors pink and black to the colors array for product 11QER/31. This product already has the black color in the list. The operation should not add duplicate values in the colors array. Only the color pink will be added.

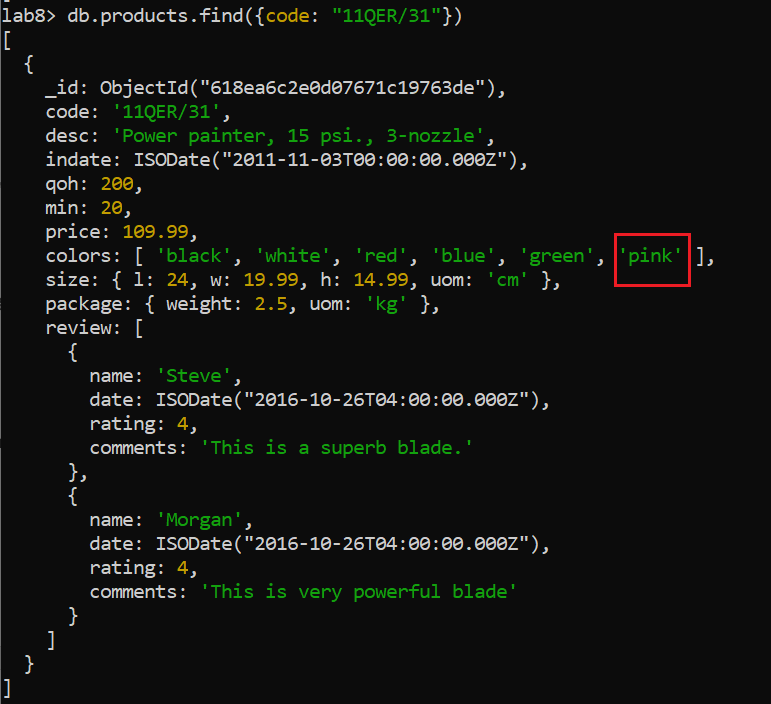
db.products.update({code: "11QER/31"},{$addToSet: {colors: {$each: ['black', 'pink']}}});



* 1. show product 11QER/31 after the update.

x`

Output:



1. **Remove elements from the end or beginning of an array.** 
   1. Remove the first element from the colors array for product 11QER/31.

db.products.update({code: "11QER/31"}, {$pop: {colors: -1}});

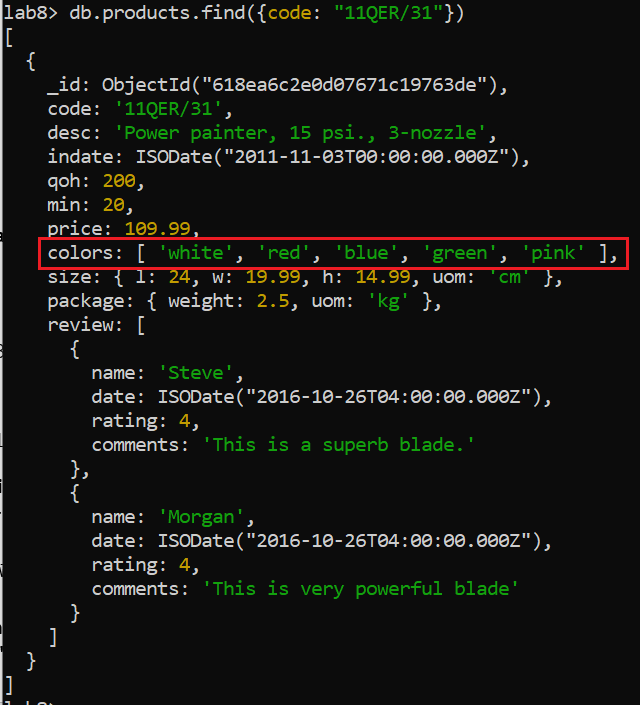


* 1. Show product 11QER/31.



Output:

Black is removed.

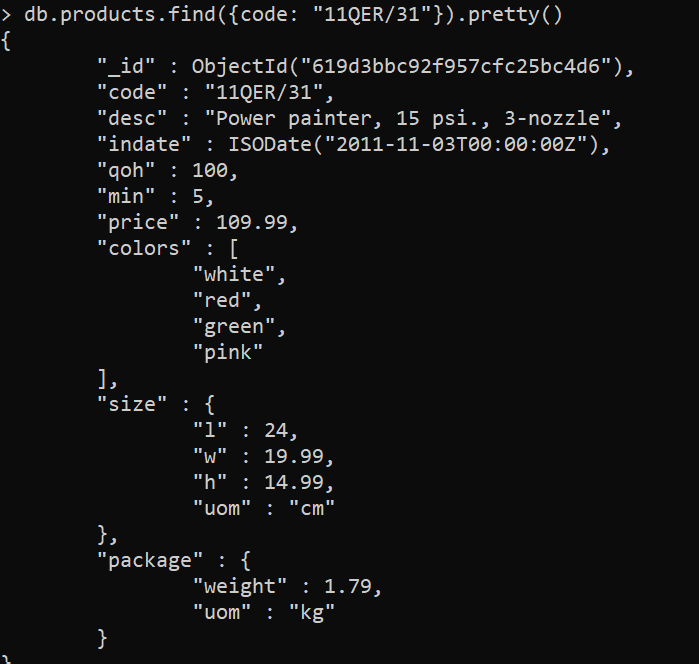


1. **Remove elements from an array based on specific criteria.** 
   1. Pull the blue color from the colors array for product 11QER/31.

db.products.update({code: '11QER/31'}, {$pull: {colors: 'blue'}})



* 1. Show product 11QER/31. Colors should not contain the element blue.

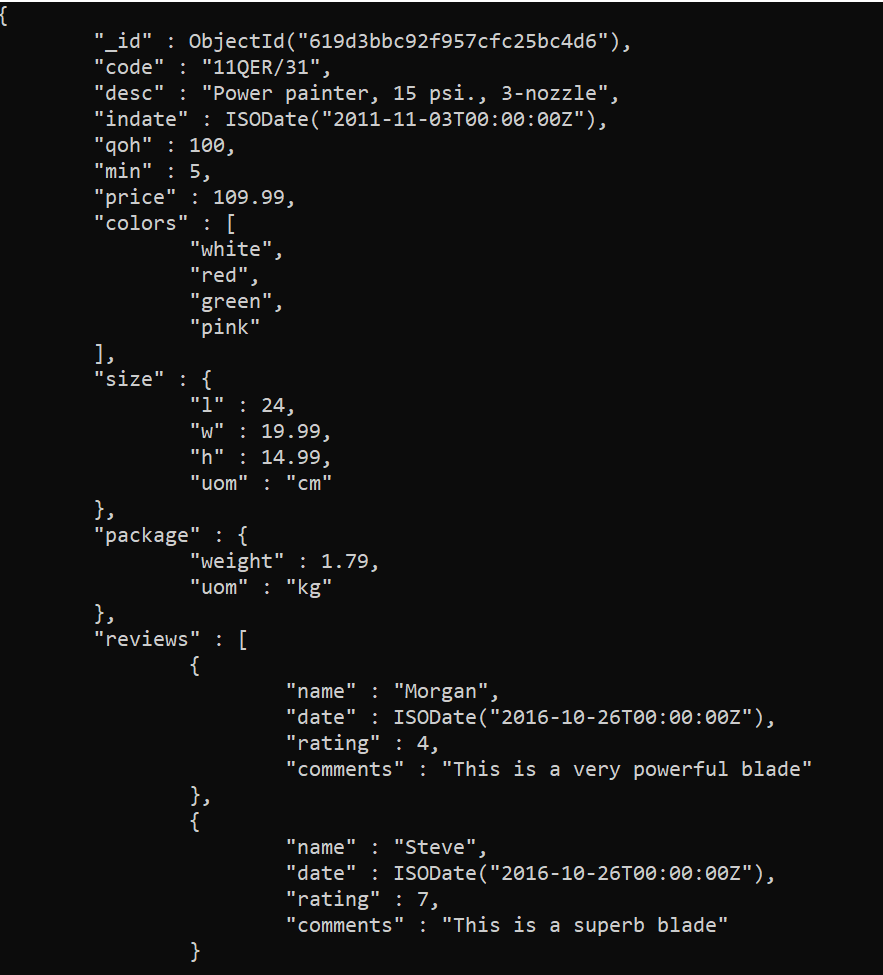


1. **Use positional operator “$” to identify which element to modify.** 
   1. Change Steve’s review rating on product 11QER/31 to 7.

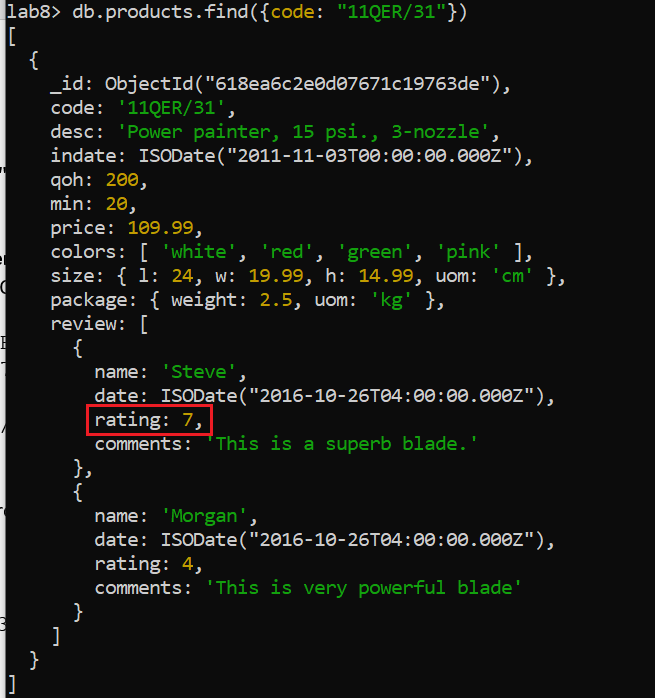
db.products.update({code: '11QER/31', 'reviews.name': 'steve'}, {$set: {'reviews.$.rating': 7}})



* 1. Show product 11QER/31 after the update.



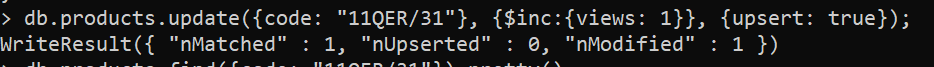
Output:



1. **Upserts: An upsert is a special type of update.**

Use $inc to increment the number of views on product 11QER/31. If the views key does not exist, it will be created.

db.products.update({code: "11QER/31"}, {$inc:{views: 1}}, {upsert: true});





Output:

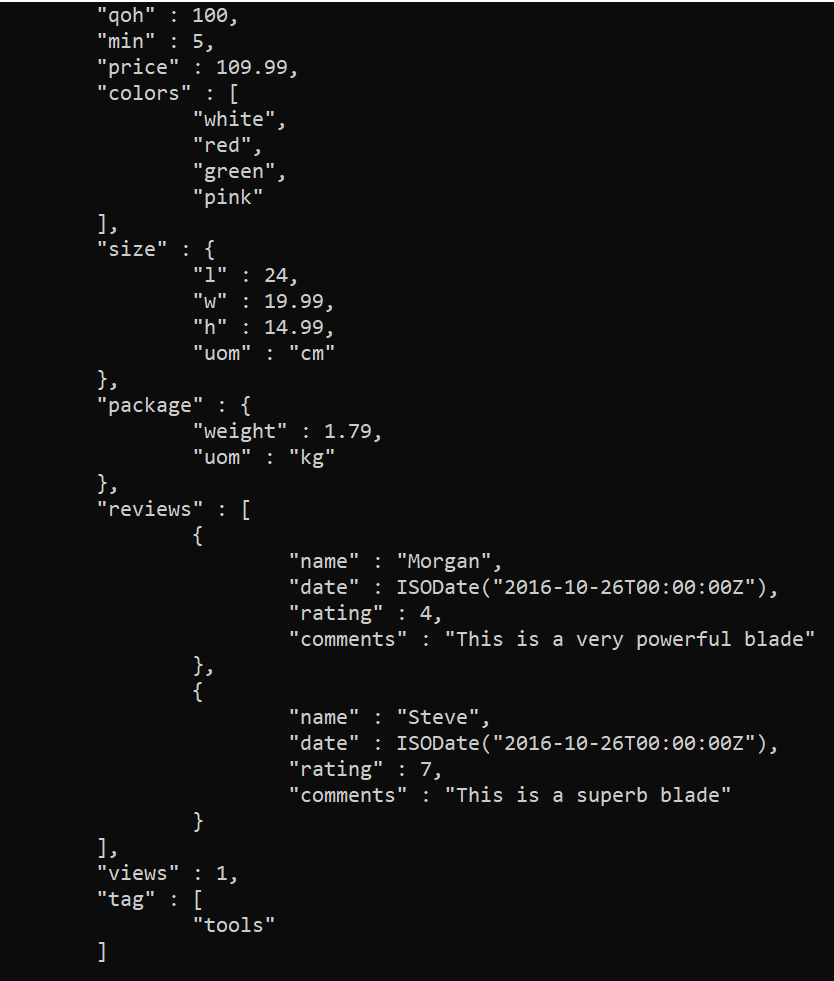


1. **Updating Multiple Documents**
   1. Use updateMany() function and the “$set” operator to roll out a new feature “tag” to all products as an array of tags. Consider the value “tools” as a tag element.

db.products.updateMany({}, {$set: {tag:['tools']}}, {$uspsert: true});

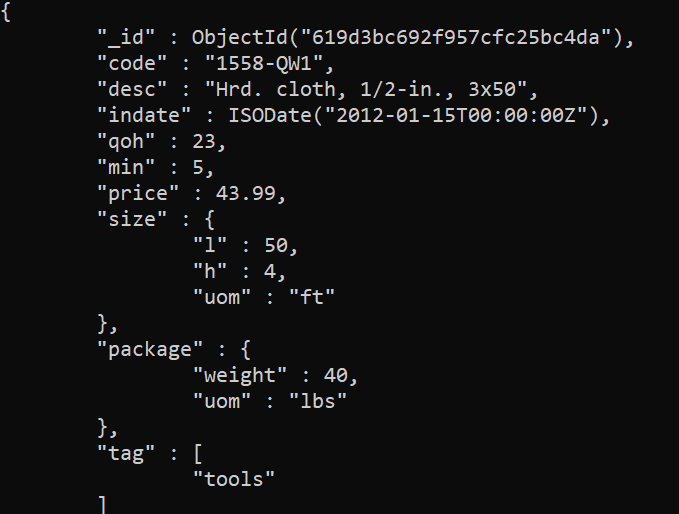


* 1. Show all the documents after the update.









Output:

All documents should have the tag key as follows:



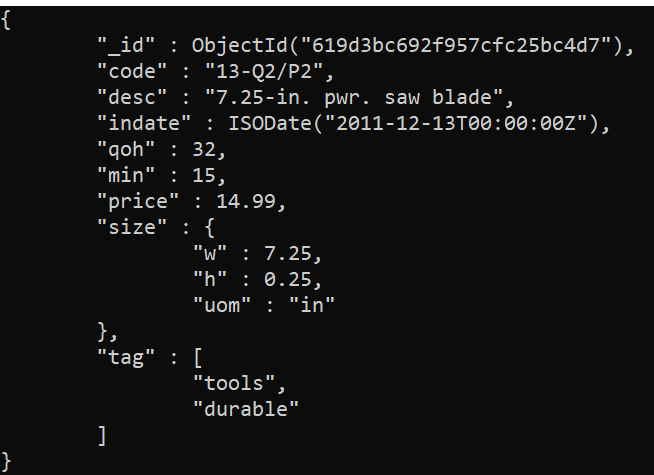
1. **Update many using the $push operator.** 
   1. Add the tag “durable” to the tag array for all products in the products collection.

db.products.updateMany({}, {$push: {tag:'durable'}});



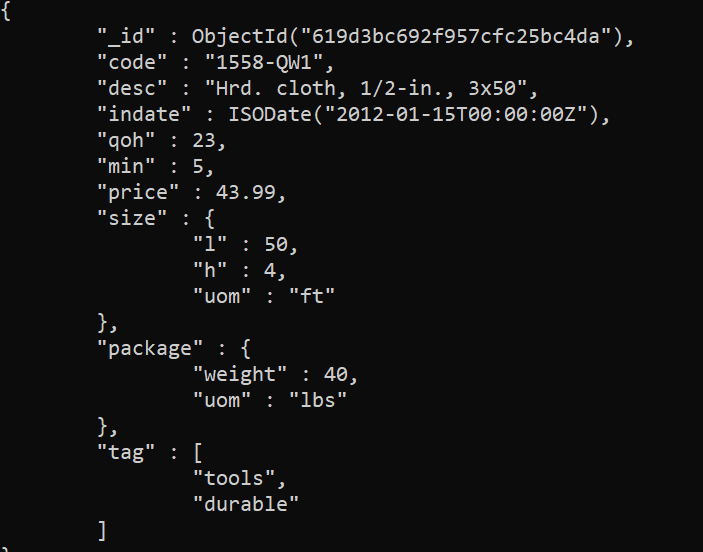
* 1. Show all the documents after the update.











Output:

All documents should have the tag key as follows:



SUBMISSION

Submit your lab8\_GroupX.doc file on BB. Replace X with your group number.

If a student does not contribute to the work, do not list his/her name(s) under the group section in the lab file and will get 0.

Submit the following files on BB:

* your lab8\_GroupX.doc: contains
  + the question as per the lab file,
  + the mongodb command **in text** and
  + a screenshot of the output, which should **NOT** contain red boxes.

Rubrics

Total: 70 pts